Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining

Project 1: Extracting Time Series Properties of Glucose Levels in Artificial Pancreas

Purpose

In this project, you will extract several performance metrics of an Artificial Pancreas system from sensor data.

Objectives

Learners will be able to:

- Extract feature data from a data set.
- Synchronize data from two sensors.
- Compute and report overall statistical measures from data.

Technology Requirements

- Python 3.10.9
- scikit-learn==1.1.1
- pandas==1.4.3
- numpy==1.23.1
- scipy==1.8.1

Project Description

In this project, we are considering the Artificial Pancreas medical control system, specifically the Medtronic 670G system. The Medtronic system consists of a continuous glucose monitor (CGM) and the Guardian Sensor (12), which is used to collect blood glucose measurements every 5 minutes. The sensor is single-use and can be used continuously for 7 days after which it has to be replaced. The replacement procedures include a recalibration process that requires the user to obtain blood glucose measurements using a Contour NextLink 2.4 glucosemeter ®.

Note that this process also requires manual intervention. The Guardian Link Transmitter®, powers the CGM sensor and sends the data to the MiniMed 670G® insulin pump. The insulin pump utilizes the Smart Guard Technology that modulates the insulin delivery based on the CGM data. The SmartGuard Technology uses a Proportional, Integrative, and Derivative controller to derive small bursts of insulin also called Micro bolus to be delivered to the user. During meals, the user uses a BolusWizard to compute the amount of food bolus required to maintain blood glucose levels. The user manually estimates the amount of carbohydrate intake and enters it to the Bolus Wizard.

The Bolus Wizard is pre-configured with the correction factor, body weight, and average insulin sensitivity of the subject, and it calculates the bolus insulin to be delivered. The user can then program the MiniMed 670G infusion pump to deliver that amount. In addition to the bolus, the MiniMed 670G insulin pump can also provide a correction bolus. The correction bolus amount is provided only if the CGM reading is above a threshold (typically 120 mg/dL) and is a proportional amount with respect to the difference of the CGM reading and the threshold.

The SmartGuard technology has two methods of suspending insulin delivery: a) Suspend on low, where the insulin delivery is stopped when the CGM reading is less than a certain threshold, or b) suspend on predicted low, where the insulin delivery is stopped when the CGM reading is predicted to be less than a certain threshold. Apart from these options, insulin delivery can also be suspended manually by the user or can be suspended when the insulin reservoir is running low.

Directions

Dataset:

You will be given two datasets:

1. From the Continuous Glucose Sensor (CGMData.csv) and
2. from the insulin pump (InsulinData.csv)

The output of the CGM sensor consists of three columns:

1. **Data time stamp (Columns B and C combined)**,
2. **the 5 minute filtered CGM reading in mg/dL**, (Column AE) and
3. the ISIG value which is the raw sensor output every 5 mins.

The output of the pump has the following information:

1. **Data time stamp**,
2. Basal setting,
3. Micro bolus every 5 mins,
4. Meal intake amount in terms of grams of carbohydrate,

5. Meal bolus,
6. correction bolus,
7. correction factor,
8. CGM calibration or insulin reservoir-related alarms, and
9. **auto mode exit events and unique codes representing reasons (Column Q)**.

The bold items are the columns that you will be using in this assignment.

Metrics to be extracted:

1. Percentage time in hyperglycemia (CGM > 180 mg/dL),
2. percentage of time in hyperglycemia critical (CGM > 250 mg/dL),
3. percentage time in range (CGM >= 70 mg/dL and CGM <= 180 mg/dL),
4. percentage time in range secondary (CGM >= 70 mg/dL and CGM <= 150 mg/dL),
5. percentage time in hypoglycemia level 1 (CGM < 70 mg/dL), and
6. percentage time in hypoglycemia level 2 (CGM < 54 mg/dL).

Each of the above-mentioned metrics are extracted in three different time intervals: daytime (6 am to midnight), overnight (midnight to 6 am), and whole day (12 am to 12 am).

Percentage is with respect to the total number of CGM data that should be available each day. Assume that the total number of CGM data that should be available is 288. There will be days such that the number of data available is less than 288, but still consider the percentage to be with respect to 288.

You have to extract these metrics for each day and then report the mean value of each metric over all days. Hence there are 18 metrics to be extracted.

The metrics will be computed for two cases:

- Case A: Manual mode
- Case B: Auto mode

Analysis Procedure:

The data is in reverse order of time. This means that the first row is the end of the data collection whereas the last row is the beginning of the data collection. The data starts with manual mode. Manual mode continues until you get a message “AUTO MODE ACTIVE PLGM OFF” in the column “Q” of the InsulinData.csv. From then onwards Auto mode starts. *You may get multiple* “AUTO MODE ACTIVE PLGM OFF” *in column “Q” but only use the earliest one to determine when you switch to auto mode. There is no switching back to manual mode,* so the first task is to determine the time stamp when Auto mode starts. **Remember that the time stamp of the CGM data is not the same**

as the timestamp of the insulin pump data because these are two different devices which operate asynchronously.

Once you determine the start of Auto Mode from InsulinData.csv, you have to figure out the timestamp in CGMData.csv where Auto mode starts. This can be done simply by searching for the time stamp that is nearest to (and later than) the Auto mode start time stamp obtained from InsulinData.csv.

For each user, CGM data is first parsed and divided into segments, where each segment corresponds to a day worth of data. One day is considered to start at 12 am and end at 11:59 pm. If there is no CGM data loss, then there should be 288 samples in each segment. The segment as a whole is used to compute the metrics for the whole day time period. Each segment is then divided into two sub-segments: daytime sub-segment and overnight subsegment. For each subsegment, the CGM series is investigated to count the number of samples that belong to the ranges specified in the metrics. To compute the percentage with respect to 24 hours, the total number of samples in the specified range is divided by 288.

Note that here you have to tackle the “missing data problem”, so a particular may not have all 288 data points. In the data files, those are represented as NaN. You need to devise a strategy to tackle the missing data problem. Popular strategies include deletion of the entire day of data, or interpolation.

Write a Python script that accepts two CSV files: CGMData.csv and InsulinData.csv and runs the analysis procedure and outputs the metrics discussed in the metrics section in another CSV file using the format described in Result.csv.

Submission Directions for Project Deliverables

Submit your Python script(main.py) and Requirements.txt file in a zip file in Coursera.

Deliverables:

- In the code you should have one “main.py” python file which the autograder can run and generate a Result.csv using the format specified.
- Assume that CGMData.csv and InsulinData.csv are already in the execution folder.
- You can have as many auxiliary python files as you want but the autograder will only run the main.py and it should generate the Result.csv. The generated Result.csv files shouldn’t include any headers and should only contain numbers in a 2 x 18 matrix.

- Requirements file (detailing how the requirements were fulfilled). See attached Requirements file template for an example. Save the Requirements file as a "Requirements.txt" file.

Note:

- While generating the Result.csv file in main.py, assume the file should be generated in the root directory to be picked up by the autograder.
- Don't add your absolute or relative path in "main.py" while submitting your solution file.

Submission Guidelines:

- Please submit a zipped file containing “main.py” and “Requirements.txt” as “**yourfirstname_lastname_Project1.zip**".
    - Do not create an additional folder; just zip the files directly.
- The submission space is located at the bottom of Week 2 under “Week 2: Graded Coursework” as “**Project Assignment: Project 1: Extracting Time Series Properties of Glucose Levels in Artificial Pancreas Submission**”.

Evaluation

The autograder in Coursera will run your Python code on a new CGMData.csv and InsulinData.csv from another subject and generate a Result.csv file. We will have a baseline Result.csv file generated from code on our end. We will match the results of your code and our code. A mean square error limit of 10% will be set as the baseline threshold for getting a full grade.

- Successful execution of your code will result in 70 points
- The remaining 30 points will be based on the total number of metrics you got within the 10% error limit.

Common Errors:

- **ModuleNotFoundError**: You are trying to access or use modules that are not found in the grader. Use the modules mentioned in the Technology Requirements section.